OIPE JUL 02 2004 PATENT & TRADEMARK OFFICE

Docket No. RSW9-2000-0090-US1 *PATENT*

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Cuomo et al.**	§	
	§	Group Art Unit: **2143**
Serial No. **09/627,518**	§	
	§	Examiner: **Mauro, Jr., Thomas J.**
Filed: **July 28, 2000**	§	
	§	
For: **Method and Apparatus for Affinity of Users to Application Servers**	§	

RECEIVED
JUL 0 8 2004
Technology Center 2100

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

ATTENTION: Board of Patent Appeals and Interferences

Certificate of Mailing Under 37 C.F.R. § 1.8(a)
I hereby certify this correspondence is being deposited with the United States Postal Service as First Class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on June 29, 2004.

By: Michele Morrow
Michele Morrow

APPELLANT'S BRIEF (37 C.F.R. 1.192)

This brief is in furtherance of the Notice of Appeal, filed in this case on April 29, 2004.

The fees required under § 1.17(c), and any required petition for extension of time for filing this brief and fees therefore, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief is transmitted in triplicate. (37 C.F.R. 1.192(a))

07/07/2004 TBESHAH1 00000021 090461 09627518
01 FC:1402 330.00 DA

OIPE JC182 JUL 02 2004 PATENT & TRADEMARK OFFICE

REAL PARTIES IN INTEREST

The real party in interest in this appeal is the following party: International Business Machines Corporation.

RECEIVED
JUL 0 8 2004
Technology Center 2100

RELATED APPEALS AND INTERFERENCES

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: 1-5, 7-15, 17-20, 25 and 26.

B. STATUS OF ALL THE CLAIMS IN APPLICATION

1. Claims canceled: 6, 16, 21-24
2. Claims withdrawn from consideration but not canceled: NONE
3. Claims pending: 1-5, 7-15, 17-20, 25 and 26
4. Claims allowed: NONE
5. Claims rejected: 1-5, 7-15, 17-20, 25 and 26

C. CLAIMS ON APPEAL

The claims on appeal are: 1-5, 7-15, 17-20, 25 and 26.

STATUS OF AMENDMENTS

Amendments made after the Final Office Action were to claims 1, 8, 11, 18, 25 and 26 for minor informalities. A Response to the Final Office Action was filed on March 30, 2004. The Advisory Action issued April 7, 2004, stated that the amendments would be entered.

SUMMARY OF INVENTION

The present invention provides a plurality of application servers, which share a database through a shared data mechanism. See page 13, lines 13-16. The application servers store session data in the shared data mechanism; therefore, if a subsequent request is routed to a different application server, the session data is available through the shared data mechanism. See page 13, lines 4-9. One or more web servers perform routing of requests to the application server. See page 13, lines 17-21. When a request is received that is accompanied by a session ID, routing is performed by utilizing a hash function on the session ID. See page 13, lines 21-23. The resulting hash value is mapped to an application server. See page 14, lines 21-28. A hash function on a session ID will always result in the same hash value; therefore, the request will always be routed to the same application server. See page 14, lines 14-17. However, if an application server is non-functional, a new hash based on the previous hash is computed until a functional application server is selected. See page 15, lines 14-21.

ISSUES

The issues on appeal are whether claims 1, 2, 5, 7, 11, 12, 15, 17 and 25 are properly rejected under 35 U.S.C. § 103(a) as being unpatentable over Chung et al. (U.S. Patent No. 6,470,389) in view of Khuc (U.S. Patent No. 6,470,008) and Johnson et al. (U.S. Patent No. 6,591,250); whether claims 3, 4, 13 and 14 are properly rejected under 35 U.S.C. § 103(a) as being unpatentable over Chung et al. (U.S. Patent No. 6,470,389), Khuc (U.S. Patent No. 6,470,008) and Johnson et al. (U.S. Patent No. 6,591,250), as applied to claims 2, 3, 12 and 13, respectively, in view of Muller et al. (U.S. Patent No. 6,606,301); and whether claims 8-10, 18-20 and 26 are properly rejected under 35 U.S.C. § 103(a) as being unpatentable over Chung et al. (U.S. Patent No.

6,470,389) in view of Khuc (U.S. Patent No. 6,470,008), Johnson et al. (U.S. Patent No. 6,591,250) and Muller et al. (U.S. Patent No. 6,606,301).

GROUPING OF CLAIMS

The claims stand or fall together as a single group.

ARGUMENT

The Final Office Action rejects claims 1, 2, 5, 7, 11, 12, 15, 17 and 25 under 35 U.S.C. § 103(a) as being unpatentable over Chung et al. (U.S. Patent No. 6,470,389) in view of Khuc (U.S. Patent No. 6,470,008) and Johnson et al. (U.S. Patent No. 6,591,250). The Final Office Action rejects claims 3, 4, 13 and 14 under 35 U.S.C. § 103(a) as being unpatentable over Chung et al. (U.S. Patent No. 6,470,389), Khuc (U.S. Patent No. 6,470,008) and Johnson et al. (U.S. Patent No. 6,591,250), as applied to claims 2, 3, 12 and 13, respectively, in view of Muller et al. (U.S. Patent No. 6,606,301). The Final Office Action rejects claims 8-10, 18-20 and 26 under 35 U.S.C. § 103(a) as being unpatentable over Chung et al. (U.S. Patent No. 6,470,389) in view of Khuc (U.S. Patent No. 6,470,008), Johnson et al. (U.S. Patent No. 6,591,250) and Muller et al. (U.S. Patent No. 6,606,301).

Chung is directed to hosting a network service on a cluster of servers, each including a primary and a secondary Internet Protocol (IP) address. In the Chung system, a common cluster address is assigned as the secondary address to each of the servers in the cluster. The cluster address may be assigned in UNIX-based servers using an ifconfig alias option, and may be a ghost IP address that is not used as a primary address by any server in the cluster. Client requests directed to the cluster address are dispatched such that only one of the servers of the cluster responds to a given client request. The dispatching may use a routing-based technique, in which all client requests directed to the cluster address are routed to a dispatcher connected to the local network of the server cluster. The dispatcher then applies a hash function to the client IP address in order to select one of the servers to process the request. The dispatching may alternatively use a broadcast-based technique, in which a router broadcasts client requests having the cluster address to all of the servers of the cluster over a local network. The servers then each provide a filtering routine, which

may involve comparing a server identifier with a hash value generated from a client address, in order to ensure that only one server responds to each request broadcast by the router.

Khuc is directed to a routing system that provides Internet service providers with Internet routing support for Internet communications. In the Khuc system, the routing system receives a query from an Internet service provider and responds with the appropriate Internet address to transport communications over the Internet. The Internet address may direct the communications to an Internet gateway or to the routing system itself.

Johnson is directed to managing virtual property by providing encryption. In the Johnson system, virtual items are each represented by one or more digital objects and are managed by one or more computer systems functioning as an owner, broker, authenticator and provider.

Muller is directed to a high performance network interface that receives network traffic in the form of packets. In the Muller system, a packet is stored in a packet queue, prior to being transferred to a host computer. Muller provides for randomly discarding a packet if the rate of packet transfers cannot keep pace with the rate of packet arrivals at the queue.

Claim 1, which is representative of the other rejected independent claims 11 and 25 with regard to similarly recited subject matter, reads as follows:

> 1. A method in a data processing system for managing a request including a session identification, comprising:
>
> calculating a first value based on the session identification;
>
> routing the request to a first server based on the first value;
>
> determining whether the first server is functional;
>
> calculating a second value based on the first value in response to the first server being non-functional; and
>
> routing the request to a second server based on the second value.

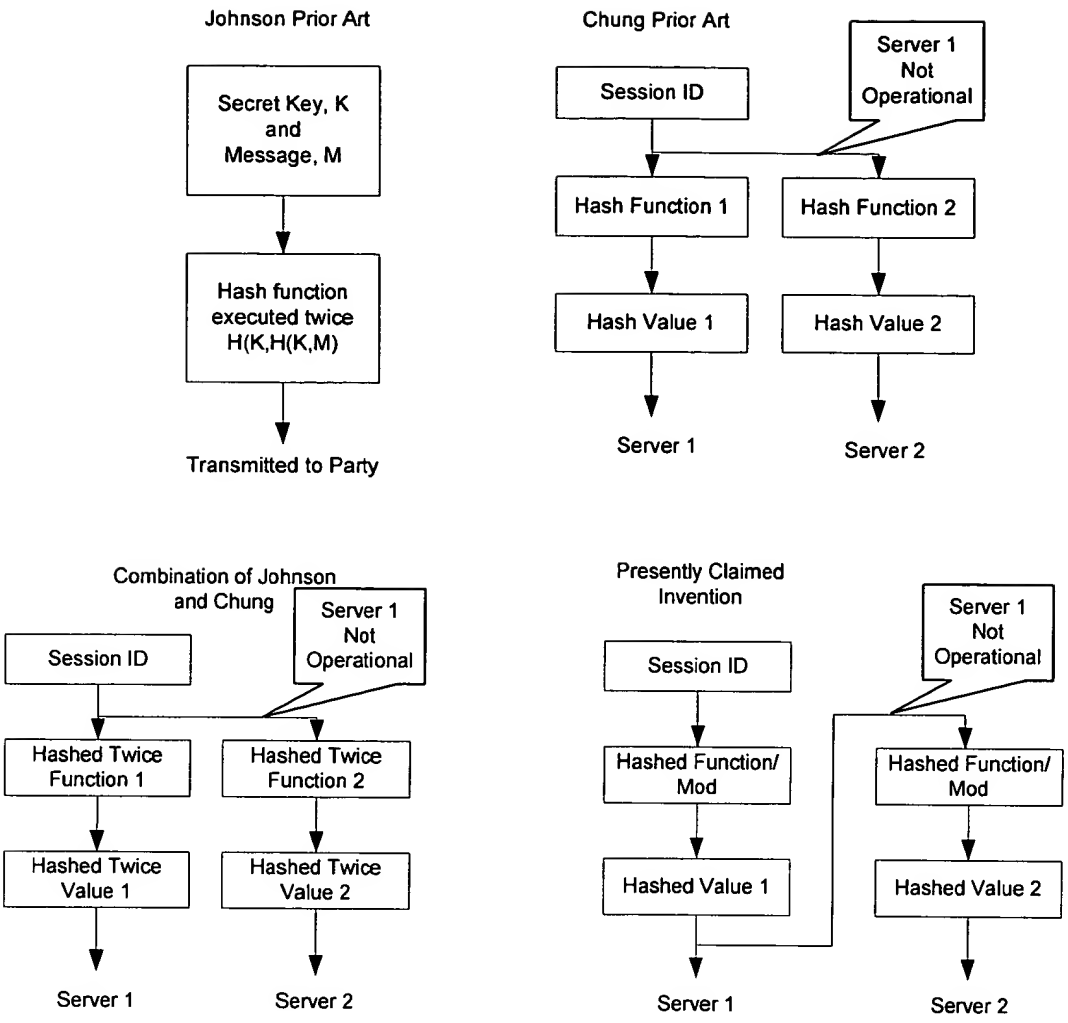
Appellants respectfully submit that the combination of Chung and Johnson, taken alone or in combination, fails to fairly teach or suggest calculating a second value based on the first value in response to the first server being non-functional.

In response, the Final Office Action, on pages 22-23, states:

> The combination of Chung and Johnson would be obvious to a person of ordinary skill in the art at the time the invention was made. Chung, as a whole, discloses a method of routing requests through a dispatcher to various servers. Server side components in his system include a router, dispatcher and the various servers comprising a server farm **[Chung – Figure 4]**. Chung teaches that in the event that a server us non-functional, i.e. a server has failed, a second hash is

> calculated in order to route the request to a different server **[Chung – Col. 7 lines 9-12]**. As stated in the above rejection, upon which the Examiner draws the applicant's attention, Johnson teaches the deficiency in the Chung reference that a second hash value is calculated from the first hash value **[Johnson – Col. 13 lines 30-32]**. While there can be various reasons why a server has failed, one all too common reason stems from an outside attack or virus that has occurred, which has left one or more servers non-functional. Therefore, Chung would see the need for added security on the server side components of his invention in order to prevent attacks from occurring. Johnson teaches that a second hash value is calculated from the first hash value in order to increase the security with which information is transmitted or stored. With this knowledge, Chung would see that by calculating a second hash on the already hashed value would provide further security in that the new server upon which being routed (a functional server) would be further disguised. This would provide added protection and prevent any other servers from being non-functional in the event of another staged attack. Therefore, the Examiner accordingly demurs to this assertion because the combination would be obvious to one of ordinary skill in the art for the reasons stated in the above office action.

Appellants respectfully disagree that Chung and Johnson, taken alone or in combination, fairly teach or suggest calculating a second value based on the first value in response to the first server being non-functional. The Examiner's arguments rely on Johnson as teaching calculating a second value based on the first value. While the Johnson reference may teach performing a hash function twice, the hash function is not performed in response to the first server being non-functional. That is, the first hashed value of Johnson is immediately hashed again to create a twice-hashed value for the purpose of increasing security. The Examiner's arguments further rely on the combination of Chung and Johnson to teach calculating a second value based on the first value in response to the first server being non-functional. Thus, Appellants submit the following diagram to depict the differences of the prior art, the combination of the prior art, and the presently claimed invention.



Thus, the combination of the Johnson and Chung references only teach a process where the session ID is hashed twice prior to sending the value to the server and if the server is non-functional, hashing another session ID twice prior to sending the values of the second server. The flowchart depicted above with regard to Chung is supported in the Chung reference at column 7, lines 9-12, which reads as follows:

> If the hash value of a given client IP address maps to the failed server, the client IP address is rehashed to map to a non-failed server, and the connections of the remaining clients are not affected by the failure.

Additionally, the flowchart depicted above with regard to Johnson is supported in the Johnson reference at column 13, lines 15-32, which reads as follows:

> Different MAC types may be used depending on the amount of security that is desired. The method may be employed as follows:

> The two communicating parties already share a secret key K, a random value that may be, for example, 16 bytes.
>
> The key is prepended to the message M to be sent, which is represented as K,M.
>
> A hash function H is executed on the key and message data together H(K,M) to produce a digest D.
>
> The digest D is used as a MAC by appending it to the message and the result M,D is transmitted to the other party.
>
> There are certain types of cryptographic attack that can theoretically compromise this type of authentication code, so an improved MAC may be used. For example, the following MAC construction that performs a hash function twice is more secure than a single hash function such as the function H(K,H(K,M)).

The flowchart depicted above with regard to the presently claimed invention is supported by the current specification at page 16, column 22-26, which reads as follows:

> If the server is down in step 616, the process sets n equal to h (step 620) and returns to step 610 to recompute the hash function. The web server recomputes the hash function until a functional application server is selected.

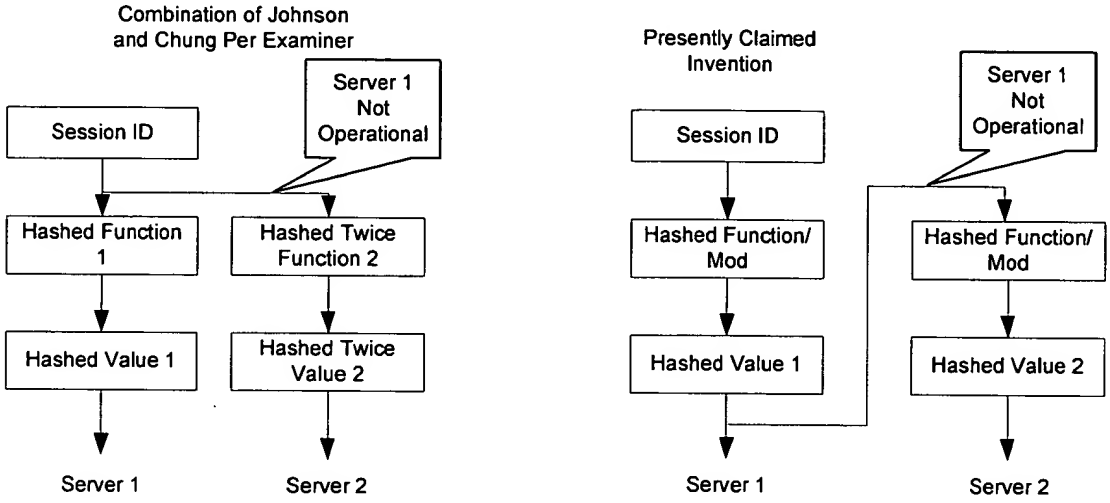
Thus, the combination of Chung and Johnson would not result in the presently claimed invention. One of ordinary skill in the art, being presented only with Chung and Johnson, and without having prior knowledge of Appellants' claimed invention, would not have found it obvious to combine and modify Chung and Johnson to arrive at Appellants' claimed invention. To the contrary, even if one were somehow motivated to combine Chung and Johnson, and it were somehow possible to combine the two systems, the result would not be the invention recited in claim 1. The result would be a process the hashes a session ID twice to create a first value and in response to the server not being available would hash the same session ID twice using another function to create a second value.

Furthermore, the Advisory Action, dated April 7, 2004, states:

> Johnson only discloses that a hash function is performed again on a value which has already been hashed. See Johnson Col. 13 lines 30-32. It is found in the Chung reference that if a first server is unavailable, i.e. non-functional, a second hash value is calculated. See Chung Col. 7, lines 9-12. The combination, as interpreted by the Examiner, brings in Johnson to rehash the already hashed value in Chung only after the first server is deemed non-functional (See final rejection), not immediately after the first hash value is calculated.

Appellants agree that Johnson performs a hash function on a value that has already been hashed; however, the first hashed value of Johnson is immediately hashed again to create a twice-hashed

value for the purpose of increasing security and performing the hash function twice is not in response to the first server being non-functional. Appellants further agree that Chung calculates a second hash function if a first server is unavailable; however, the second hash function is not calculating based on the first value from the first hash function. Thus, Appellants respectfully submit that the following diagram depicts the Examiner's interpretation.



Thus, as per the Examiner's interpretation, if the server is deemed non-operational by Chung, the same session ID will be twice-hashed by Johnson to provide a second value. The result would not be calculating a second value based on the first value in response to the first server being non-functional.

Independent claims 8, 18 and 26, recite similar subject matter to that of independent claims 1, 11 and 25. That is, independent claims 8, 18 and 26, recite "performing a hash function on the first hash value to form a second hash value in response to the first server being non-functional." Thus, independent claims 8, 18 and 26, distinguish over the combination of Chung, Johnson, Muller and Khuc for at least the reasons noted above with regard to independent claims 1, 11 and 25.

Additionally, the Final Office Action, on page 24, states:

> While applicant argues that "the same hash function: is used to calculate the second value, nowhere does this statement appear in the claims. As is stated in claim 1, "a second hash value is calculated from the first value." Therefore, the applicant's argument with regard to "the same hash function" does not commensurate with the claim language.

Whether or not the same hash function is used to calculate the second hash value is irrelevant in view of the arguments above. That is, the combination of Chung and Johnson does not result in the presently claimed invention. The result of the combination of the Johnson and Chung references would be a process that hashes a session ID twice to create a first value and in response to the server not being available would hash the same session ID twice using another function to create a second value.

Furthermore, the Final Office Action, on page 24, states:

> The combination of Chung and Muller would be obvious to a person of ordinary skill in the art at the time the invention was made. Chung, as a whole, discloses a method of routing requests through a dispatcher to various servers. In order to route the requests to the various servers with speed and efficiency, Chung hashes the IP address of the client and uses the hashed value to select a server. By hashing the value, a smaller, whole number is obtained which can quickly and efficiently be used to select a server and rout a request. Thus, Chung wishes to obtain speed and efficiency in routing, which he does obtain by using a hash function. Similarly, Muller discloses a modulus function to after he obtains a hash value to further obtain a smaller value with which to route requests to one of a number of processors, i.e. server. Muller uses the modulus function in order to obtain a smaller value which he further discloses for increased speed in routing to the various number of processors. Therefore, because both Chung and Muller are calculating values to route to different processors or server and because both are doing it to obtain speed and efficiency and to prevent "bottleneck" situations from developing, it would have been obvious to combine the two references. Chung would want to incorporate the modulus function in order to further gain speed and efficiency in addition to doing his hash function. Therefore, the Examiner accordingly demurs to this assertion because the combination would be obvious to one of ordinary skill in the art for the reasons stated in the above office action.

The Muller reference is used as a basis to reject claims 3, 4, 8-10, 13, 14, 18-20 and 26. Claims 8, 18 and 26 were addressed above in that the combination of Chung, Khuc and Johnson do not teach or suggest "performing a hash function on the first hash value to form a second hash value in response to the first server being non-functional." Claims 3, 4, 9, 10, 13, 14, 19 and 20 are dependent on claims 1, 8, 11 and 18 and, thus, are distinguished over the combination of Chung, Khuc, Johnson and Muller for at least the reasons noted above with regard to claims 1, 8, 11 and 18. Moreover, Muller does not provide for the deficiencies of Chung, Khuc and Johnson and, thus, any alleged combination of Muller with Chung, Khuc and Johnson would not be sufficient to reject claims 1, 8, 11 and 18 or claims 3, 4, 9, 10, 13, 14, 19 and 20 by virtue of their dependency. That

is, Muller does not teach calculating a second value based on the first value in response to the first server being non-functional, as recited in independent claims 1, 11 and 25 or performing a hash function on the first hash value to form a second hash value in response to the first server being non-functional, as recited in independent claims 8, 18 and 26.

Still further, the Final Office Action, on pages 24-25, states:

> The combination of Chung and Khuc would be obvious to a person of ordinary skill in the art at the time the invention was made. Chung, as a whole, discloses a method of routing requests through a dispatcher to various servers. In order to route the requests to the various servers with speed and efficiency, Chung hashes the IP address of the client and uses the hashed value to select a server. By hashing the value, a smaller, whole number is obtained which can quickly and efficiently be used to select a server and rout a request. Thus, Chung wishes to obtain speed and efficiency in routing, which he does obtain by using a hash function. Khuc, as is well known in the art, teaches the use of a look-up table to quickly and efficiently route requests. Because this technique of using a routing look-up table was well known in the art and Chung wants to gain speed and efficiency for routing, something a look-up table provides, it would have been obvious to one of ordinary skill in the art for Chung to incorporate the look-up table teachings of Khuc. Therefore, the Examiner accordingly demurs to this assertion because the combination would be obvious to one of ordinary skill in the art for the reasons stated in the above office action.

The Khuc reference is used as a basis to reject claims 1-5, 7-15, 17-20, 25 and 26. Claims 8, 18 and 26 were addressed above in that the combination of Chung, Khuc, Johnson and Muller do not teach or suggest "performing a hash function on the first hash value to form a second hash value in response to the first server being non-functional." As claims 2-5, 7, 9, 10, 12-15, 17, 19 and 20 are dependent on claims 1, 8, 11 and 18 and, thus, are distinguished over the combination of Chung, Khuc and Johnson for at least the reasons noted above with regard to claims 1, 8, 11 and 18. Moreover, Khuc does not provide for the deficiencies of Chung, Johnson or Muller and, thus, any alleged combination of Khuc with Chung, Johnson and Muller would not be sufficient to reject claims 1, 8, 11 and 18 or claims 2-5, 7, 9, 10, 12-15, 17, 19 and 20 by virtue of their dependency. That is, Khuc does not teach calculating a second value based on the first value in response to the first server being non-functional, as recited in independent claims 1, 11 and 25 or performing a hash function on the first hash value to form a second hash value in response to the first server being non-functional, as recited in independent claims 8, 18 and 26.

In view of the above, Appellants respectfully submit that Chung, Khuc, Johnson and Muller, taken alone or in combination, fail to teach or suggest all of the features of claim 1, or the similar features found in independent claims 8, 11, 18, 25 and 26. At least by virtue of their dependency on claims 1, 8, 11 and 18, the specific features of dependent claims 2-5, 7, 9, 10, 12-15, 17, 19 and 20, are not taught or fairly suggested by Chung, Khuc, Johnson and Muller, whether taken alone or in combination. Accordingly, Appellants respectfully submit that the rejection of claims 1-5, 7-15, 17-20, 25 and 26 under 35 U.S.C. § 103(a) should be overturned.

CONCLUSION

In view of the above, Appellants respectfully submit that claims 1-5, 7-15, 17-20, 25 and 26 are allowable over the cited prior art and that the application is in condition for allowance. Accordingly, Appellant respectfully requests the Board of Patent Appeals and Interferences to not sustain the rejections set forth in the Final Office Action.

Respectfully submitted,

Francis Lammes
Reg. No. 55,353
Yee & Associates, P.C.
PO Box 802333
Dallas, TX 75380
(972) 367-2001

APPENDIX OF CLAIMS

The text of the claims involved in the appeal are:

1. A method in a data processing system for managing a request including a session identification, comprising:

calculating a first value based on the session identification;

routing the request to a first server based on the first value;

determining whether the first server is functional;

calculating a second value based on the first value in response to the first server being non-functional; and

routing the request to a second server based on the second value.

2. The method of claim 1, wherein the step of calculating a first value comprises performing a hash function on the session identification.

3. The method of claim 2, wherein the step of routing the request to a first server comprises:

performing a modulus function on the first value to form a first integer; and

selecting a first server based on the first integer.

4. The method of claim 3, wherein the step of selecting a server comprises looking up the server in a look-up table using the first integer.

5. The method of claim 1, wherein the step of routing the request to a server comprises:

selecting a first server based on the first value;

determining whether the first server is functional; and

routing the request to the first server in response to the first server being functional.

7. The method of claim 5, wherein the step of determining whether the first server is functional comprises using a look-up table.

8. A method in a data processing system for routing a request to one of a number of servers, comprising:

receiving a request including a session identification;

performing a hash function on the session identification to form a first hash value;

performing a modulus function on the first hash value to form a first integer;

routing the request to a first server based on the first integer in response to the first server being functional;

performing a hash function on the first hash value to form a second hash value in response to the first server being non-functional;

performing a modulus function on the second hash value to form a second integer; and

routing the request to a second server based on the second integer.

9. The method of claim 8, wherein the integer is between zero and the number of servers minus one.

10. The method of claim 8, wherein the step of routing the request comprises looking up the server in a look-up table using the integer.

11. An apparatus for managing a request including a session identification, comprising:

first calculation means for calculating a first value based on the session identification;

first routing means for routing the request to a server based on the first value;

determining means for determining whether the first server is functional;

second calculation means for calculating a second value based on the first value in response to the first server being non-functional; and

second routing means for routing the request to a second server based on the second value.

12. The apparatus of claim 11, wherein the first calculation means comprises hash means for performing a hash function on the session identification.

13. The apparatus of claim 12, wherein the first routing means comprises:

modulus means for performing a modulus function on the first value to form a first integer; and

selection means for selecting a first server based on the first integer.

14. The apparatus of claim 13, wherein the selection means comprises table means for looking up the server in a look-up table using the first integer.

15. The apparatus of claim 11, wherein the first routing means comprises:

selection means for selecting a first server based on the first value;

determining means for determining whether the first server is functional; and

means for routing the request to the first server in response to the first server being functional.

17. The apparatus of claim 15, wherein the determining means uses a look-up table.

18. An apparatus for routing a request to one of a number of servers, comprising:

a processor; and

a memory electrically connected to the processor, the memory having stored therein a program to be executed on the processor for performing:

receiving a request including a session identification;

performing a hash function on the session identification to form a first hash value;

performing a modulus function on the first hash value to form a first integer;

routing the request to a first server based on the first integer in response to the first server being functional;

performing a hash function on the first hash value to form a second hash value in response to the first server being non-functional;

performing a modulus function on the second hash value to form a second integer; and

routing the request to a second server based on the second integer.

19. The apparatus of claim 18, wherein the integer is between zero and the number of servers minus one.

20. The apparatus of claim 18, wherein the step of routing the request comprises looking up the server in a look-up table using the integer.

25. A computer program product, in a computer readable medium, for managing a request including a session identification, comprising:

instructions for calculating a first value based on the session identification;

instructions for routing the request to a server based on the first value;

instructions for determining whether the first server is functional;

instructions for calculating a second value based on the first value in response to the first server being non-functional; and

instructions for routing the request to a second server based on the second value.

26. A computer program product, in a computer readable medium, for routing a request to one of a number of servers, comprising:

instructions for receiving a request including a session identification;

instructions for performing a hash function on the session identification to form a first hash value;

instructions for performing a modulus function on the first hash value to form a first integer;

instructions for routing the request to a first server based on the first integer in response to the first server being functional;

instructions for performing a hash function on the first hash value to form a second hash value in response to the first server being non-functional;

instructions for performing a modulus function on the second hash value to form a second

integer; and

instructions for routing the request to a second server based on the second integer.